(12) UK Patent Application (19) GB (11) 2 311 880 (13) A

(43) Date of A Publication 08.10.1997

(21) Application No 9607044.6

(22) Date of Filing 03.04.1996

(71) Applicant(s)
Advanced RISC Machines Limited

(Incorporated in the United Kingdom)

90 Fulbourn Road, Cherry Hinton, CAMBRIDGE, CB1 4JN, United Kingdom

(72) Inventor(s)
Simon Charles Watt

(74) Agent and/or Address for Service
D Young & Co
21 New Fetter Lane, LONDON, EC4A 1DA, United Kingdom

(51) INT CL^6
G06F 12/08

(52) UK CL (Edition O)
G4A AMC

(56) Documents Cited
GB 2292822 A **GB 2250114 A** **GB 2214336 A**
EP 0481616 A2 **EP 0466265 A1** **EP 0442474 A2**
EP 0075714 A2 **US 5434992 A**

(58) Field of Search
UK CL (Edition O) **G4A AMC**
INT CL^6 **G06F 12/08**
On-line: WPI, Inspec, Computer

(54) **Partitioned cache memory**

(57) A data processing system incorporating a cache memory 2 and a central processing unit 4. A storage control circuit 10 is responsive to a programmable partition setting PartVal to partition the cache memory between instruction words and data words. The central processing unit 4 indicates with a signal I/D whether the word to be stored within the cache memory 2 resulted from an instruction word cache miss or a data word cache miss. In alternative embodiments a data cache is partitioned between a main processor and a co-processor (Fig 8) or between various tasks (Fig 9).

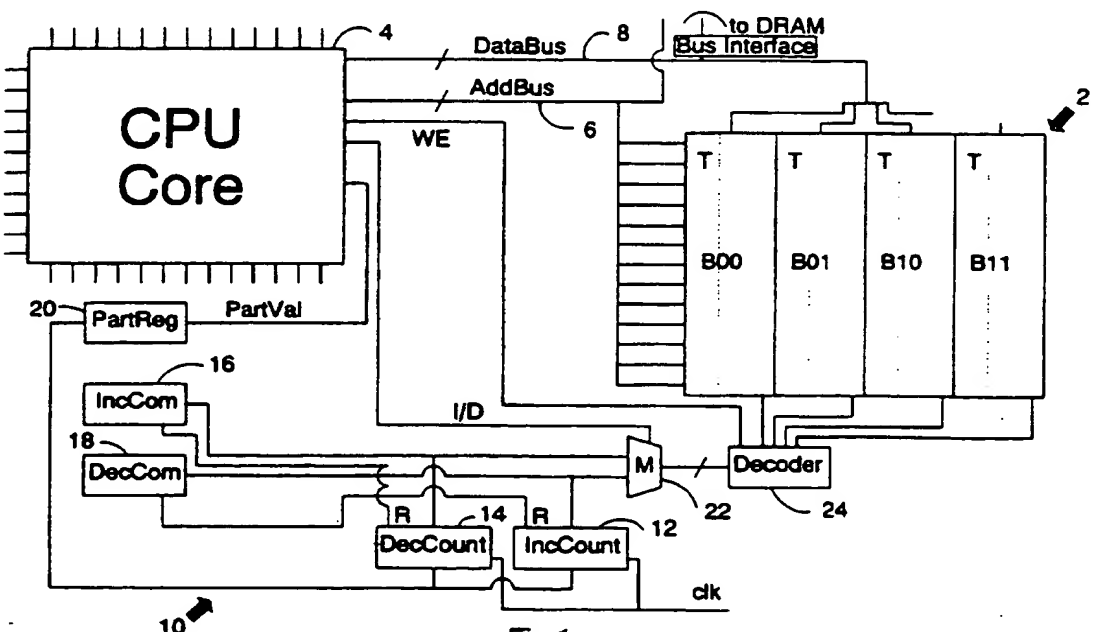
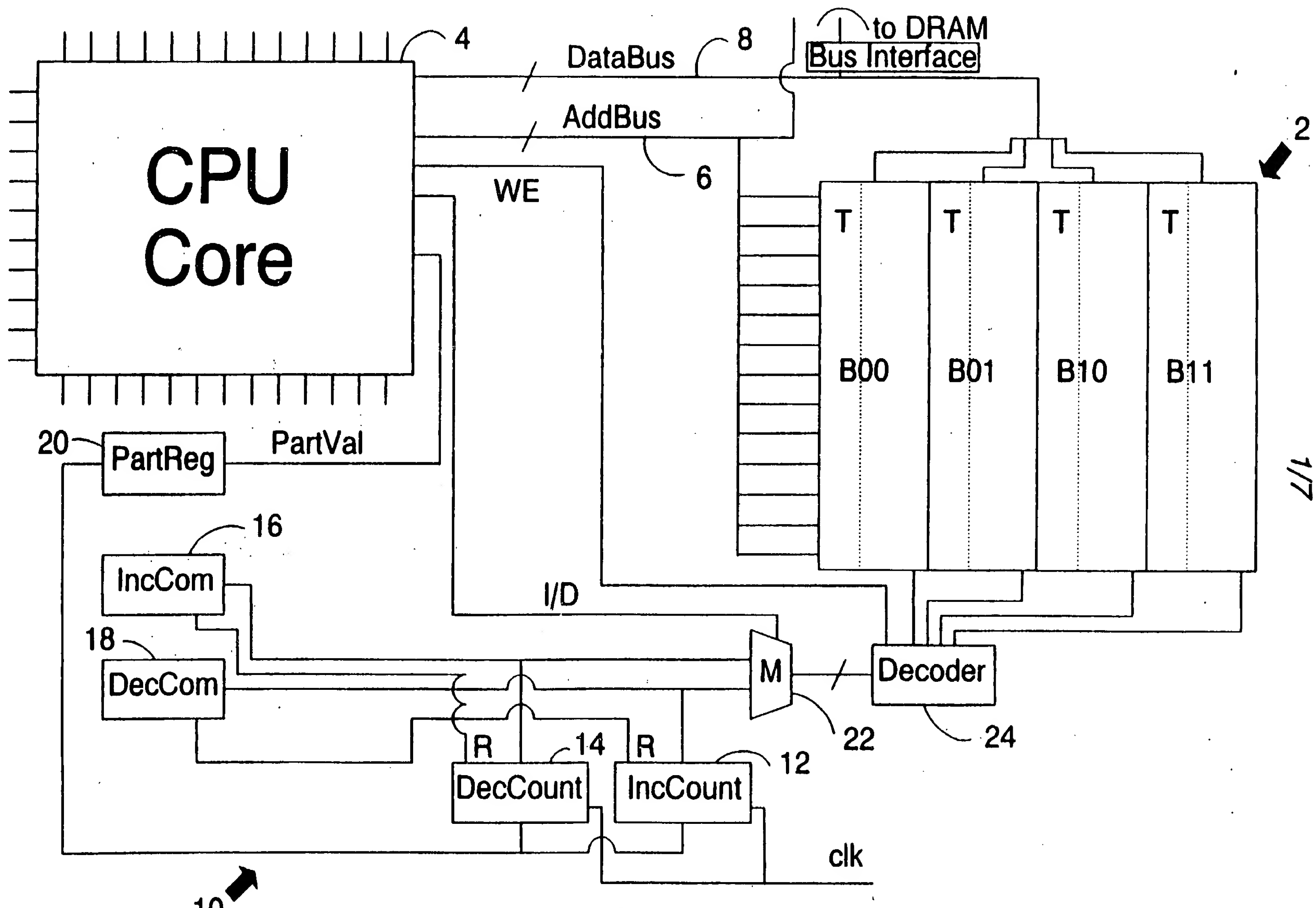


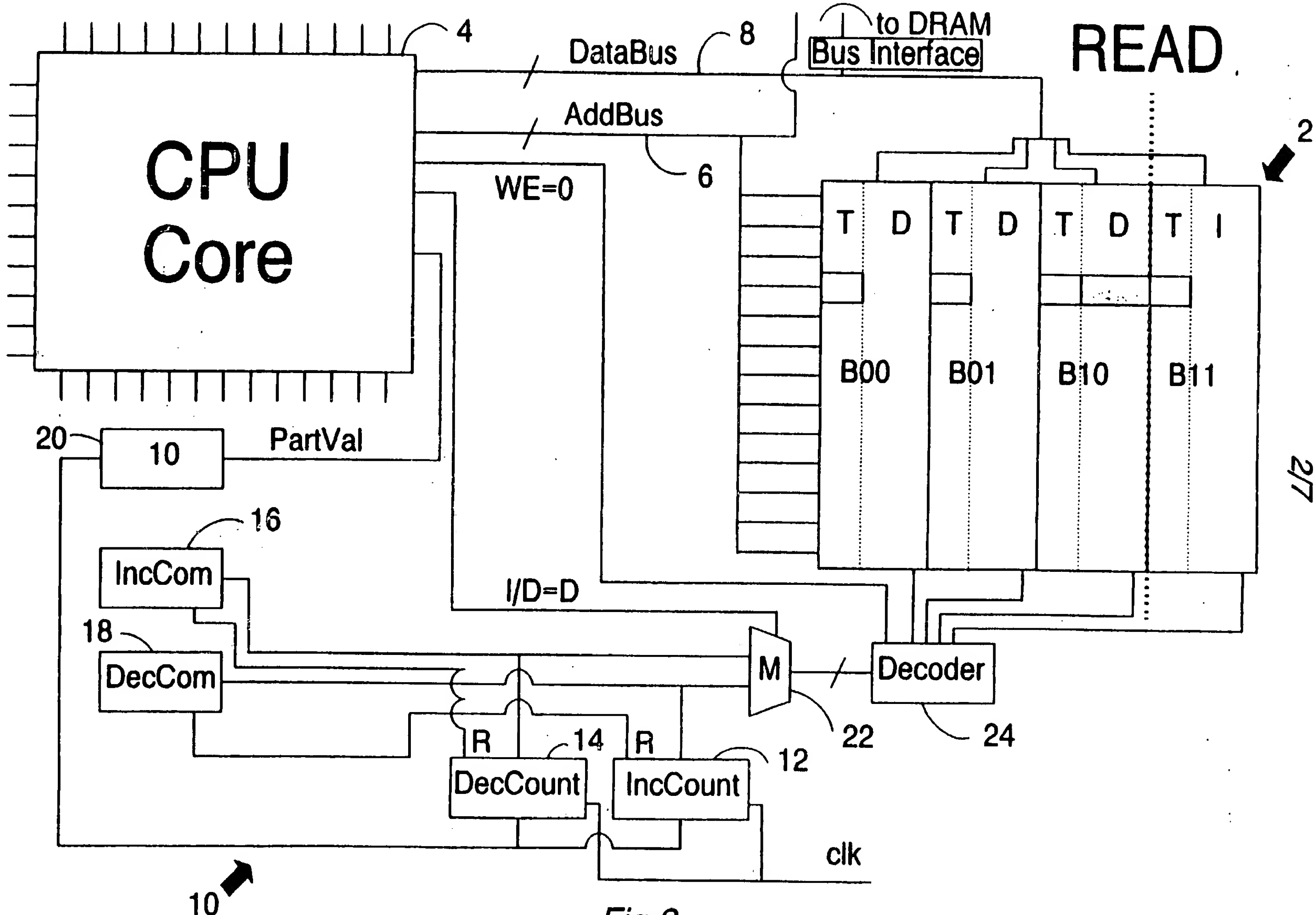
Fig. 1

GB 2 311 880 A



4
DataBus
8
to DRAM
Bus Interface
AddBus
6
2
WE
CPU Core
T
B00
B01
B10
B11
20
PartReg
PartVal
16
IncCom
I/D
18
DecCom
M
Decoder
22
24
R
14
R
12
DecCount
IncCount
clk
10

Fig.1



READ
CPU Core
4
DataBus
8
to DRAM
Bus Interface
AddBus
6
WE=0
2
T
D
T
D
T
D
T
I
B00
B01
B10
B11
20
10
PartVal
16
IncCom
18
DecCom
I/D=D
M
Decoder
22
24
R
14
R
12
DecCount
IncCount
clk
10

Fig.2

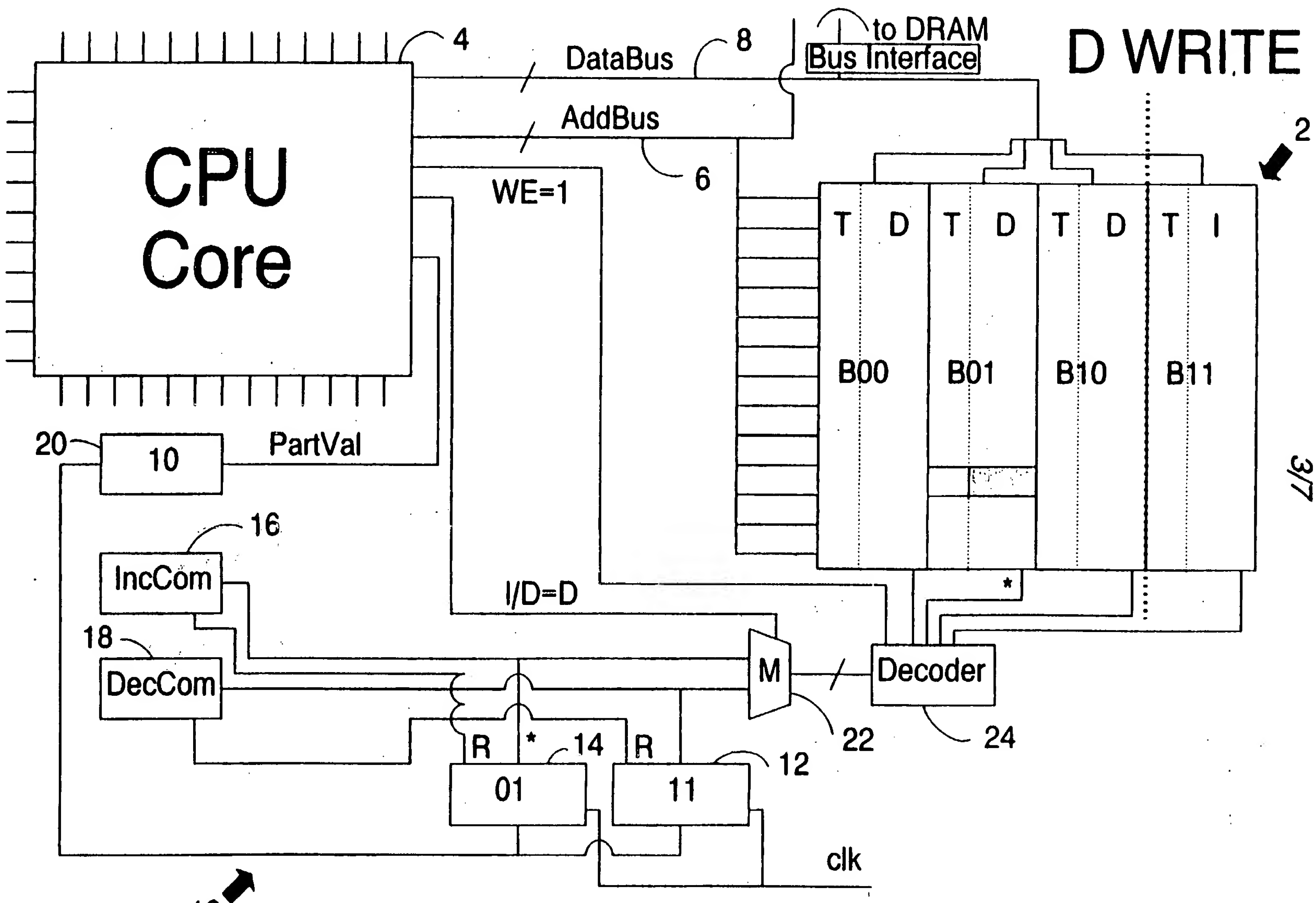
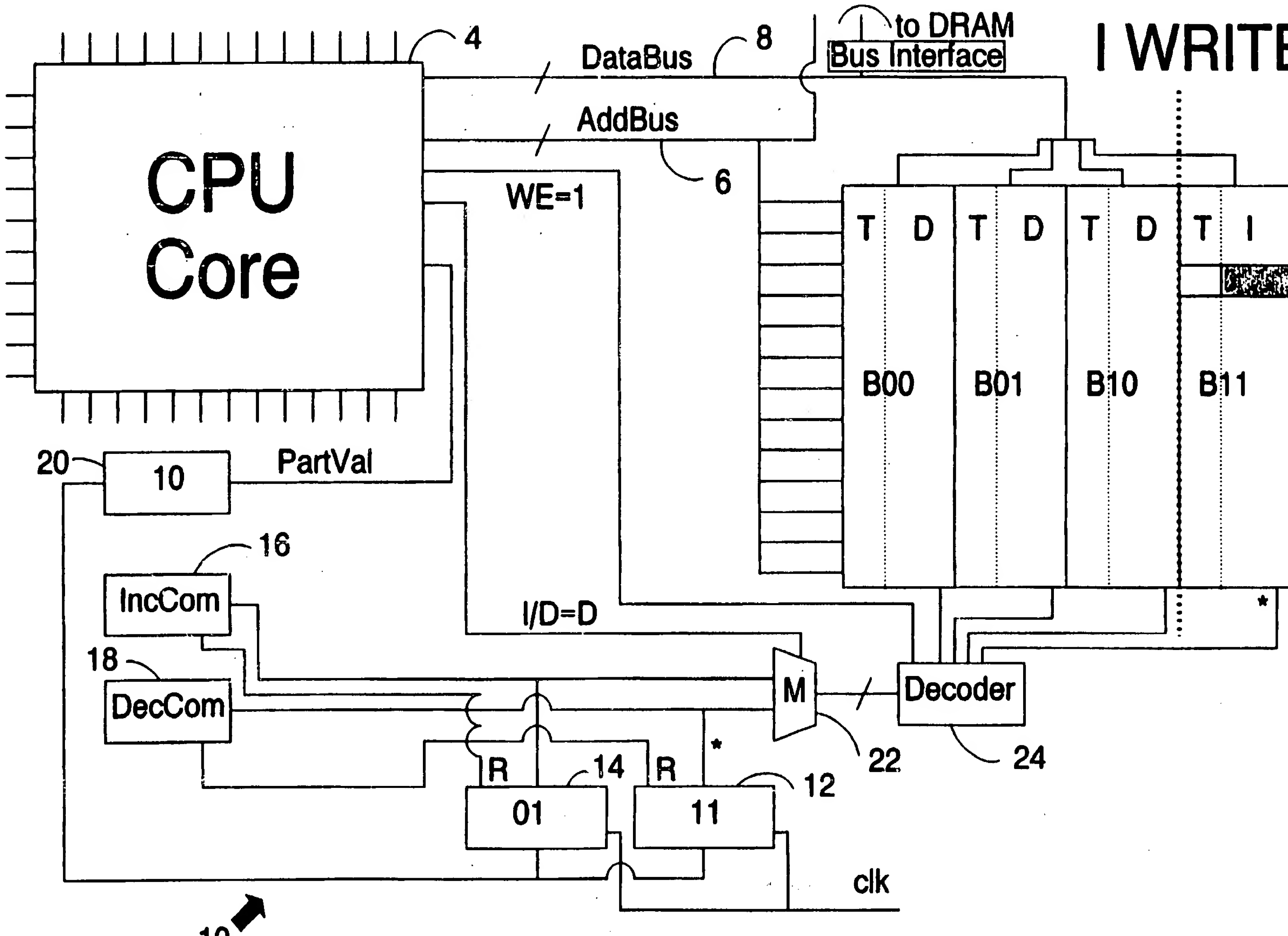

D WRITE
CPU
Core
4
DataBus
8
to DRAM
Bus Interface
AddBus
6
WE=1
2
T
D
T
D
T
D
T
I
B00
B01
B10
B11
20
10
PartVal
16
IncCom
18
DecCom
I/D=D
*
M
Decoder
22
24
R
*
14
R
12
01
11
clk
10

Fig.3

I WRITE



CPU
Core
4
DataBus
8
to DRAM
Bus Interface
AddBus
6
WE=1
2
T
D
T
D
T
D
T
I
B00
B01
B10
B11
20
10
PartVal
16
IncCom
18
DecCom
I/D=D
M
Decoder
22
24
R
14
R
12
01
11
*
clk
10

Fig.4

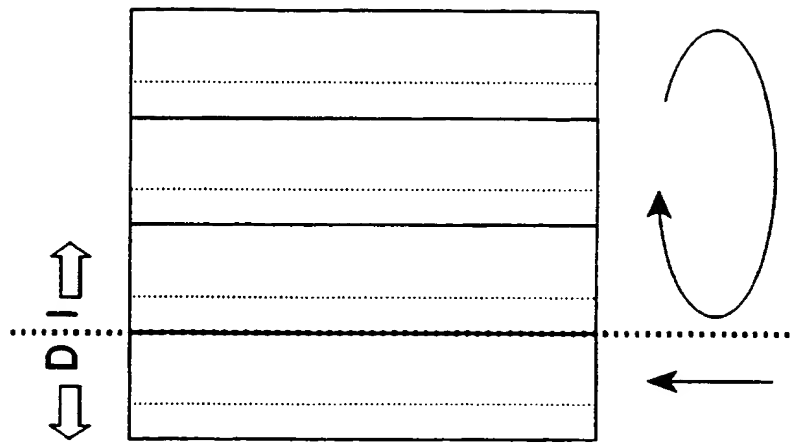


Fig.7

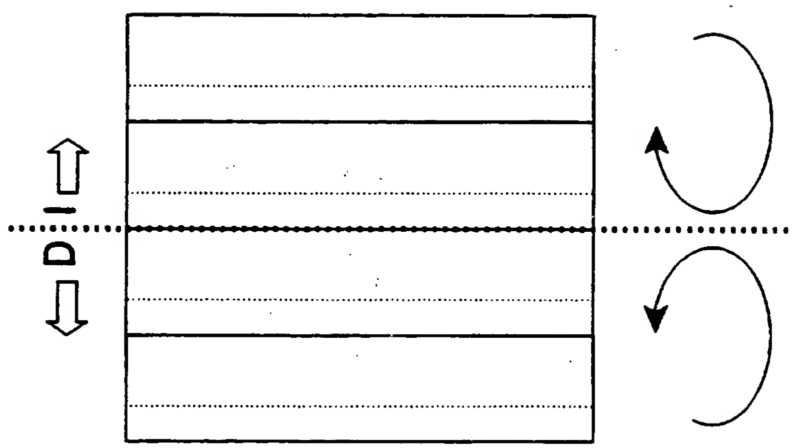


Fig.6

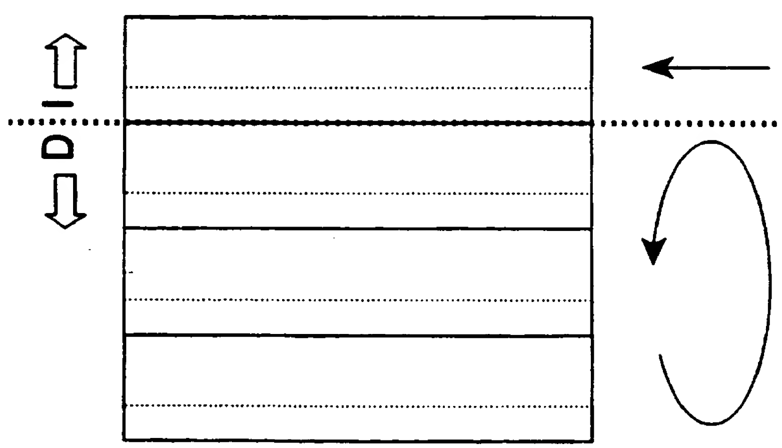


Fig.5

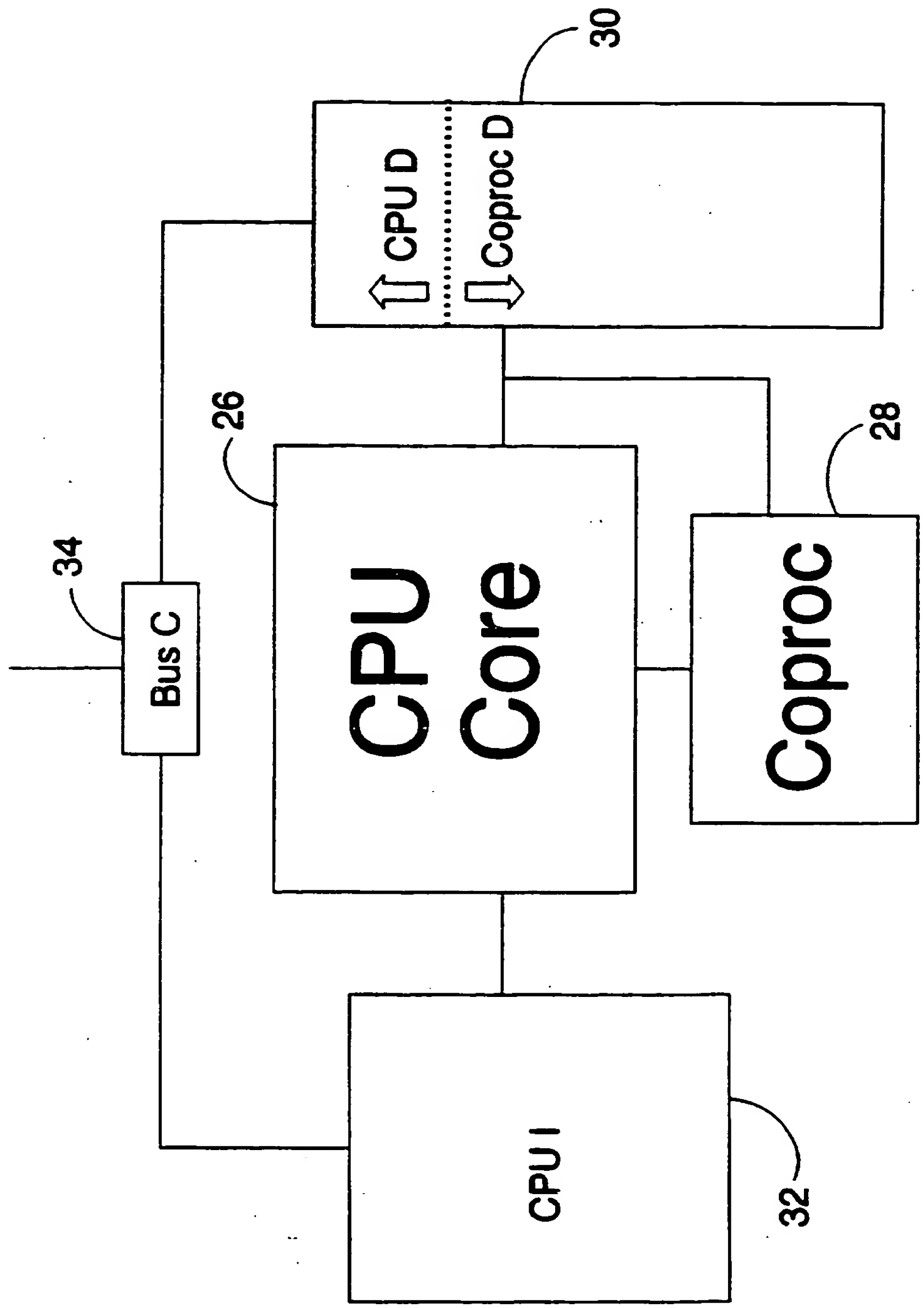


Fig.8

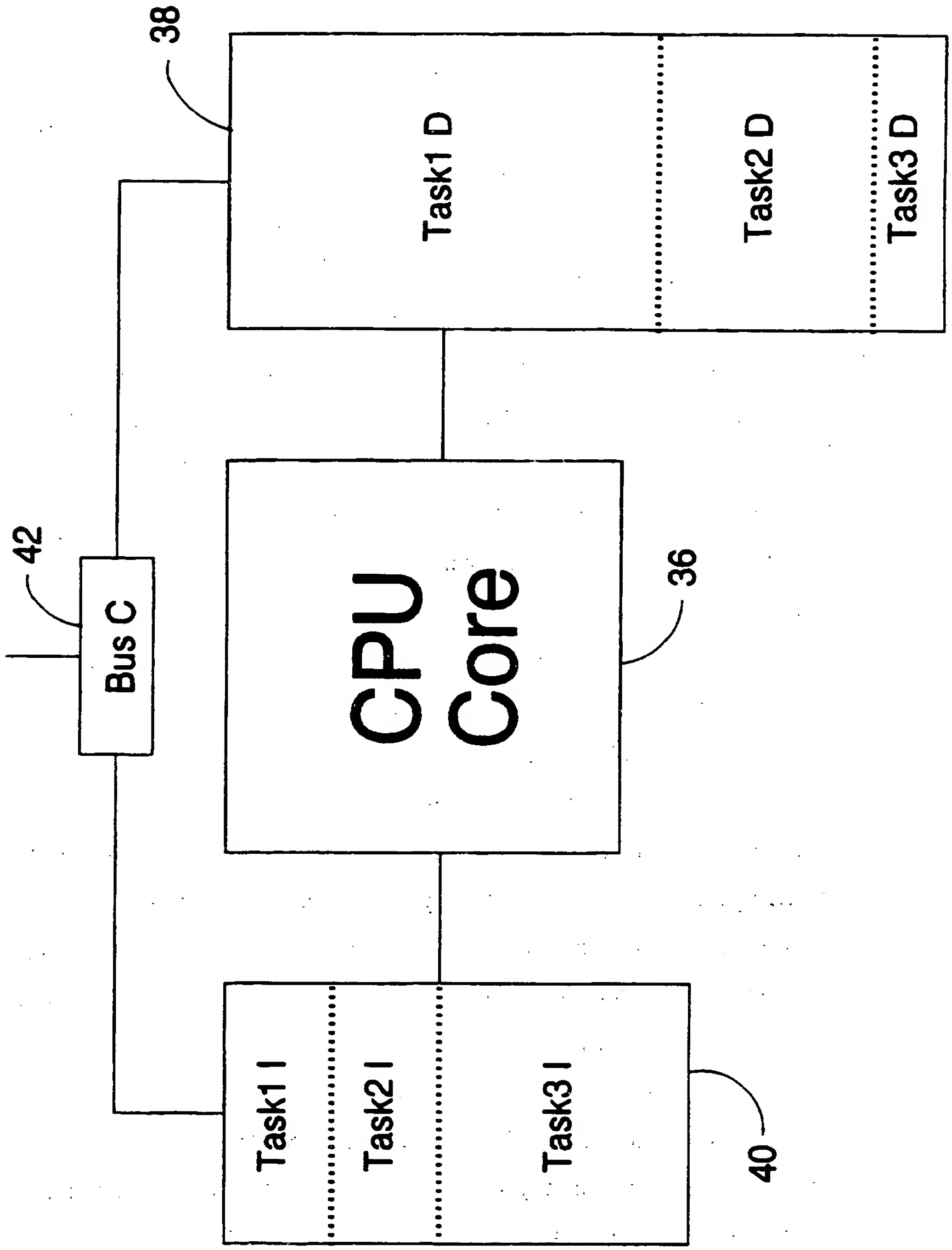

38
Task1 D
Task2 D
Task3 D
42
Bus C
CPU
Core
36
Task1 I
Task2 I
Task3 I
40

Fig.9

CACHE MEMORY CONTROL

This invention relates to the field of data processing. More particularly, this invention relates to data processing systems incorporating cache memories and the control of these cache memories.

It is known to provide data processing systems with cache memories in order to yield performance improvements through the ability to rapidly access frequently used data or instructions. There are many different types of cache architecture that may be employed. In some architectures data and instructions share a single cache. In other architectures, data and instructions have their own separate caches. Within each cache various replacement algorithms can be used to determine which words (data or instructions) should be held within the cache and which should be overwritten. Examples of these replacement algorithms would be a least recently used algorithm that discards the word that was least recently accessed or a random replacement algorithm.

As processor speeds have risen within data processing systems there has been an increasing reliance upon cache techniques to maintain system performance. In particular, if a cache miss occurs (a request to the cache for a word that is not in fact stored there), then a much slower access must be made to a different memory, such as external dynamic random access memory, which stalls the processor and has a marked impact upon the system performance. The provision of separate instruction and data caches can reduce this problem since it is possible to ensure that frequently accessed instructions, such as within a program loop, are safely cached and not likely to be overwritten with data and vice versa. Furthermore, separate instruction and data caches allow parallel accesses to these caches to be made.

Another technique for improving performance would be to increase the cache size. However, increasing the cache size has the disadvantage of increasing the cost of the system. In particular, as an integrated circuit becomes larger due to the increased area of cache memory, the production yield (fewer ICs per wafer and higher probability of a defect occurring in a given IC) decreases and so the individual price of each integrated circuit increases.

A disadvantage of separate instruction and data caches is that the division of

the total cache capacity between data and instructions is fixed with the manufacture of the system (in fact usually at the time of design many months before manufacture) and for a given processing task may be inappropriate, e.g. a particular task may require very few instructions relative to the amount of data with the result that the performance is constrained by the data cache size with the instruction cache being relatively under used. It can be very hard to predict the best split. especially for a general purpose microprocessor, and very expensive in computer time as simulations run many orders of magnitude slower than the real hardware.

An alternative cache architecture that reduces at least some of these problems is to use a single cache incorporating a lock down mechanism. In this way. individual words or areas of the cache may be loaded and then a flag set to indicate that they should remain permanently in place within the cache and not be subject to replacement when a cache miss occurs. An example would be that the instructions for a program loop could be loaded into a cache and then locked down such that those frequently accessed and performance critical items were always available in a cached form. Another example is the code for responding to a particular type of interrupt that is so performance critical that it must always be cached, even though it would not be justified on the basis of a least recently used replacement algorithm, then such interrupt instructions could be loaded into the cache and then locked down such that they were permanently present there.

A disadvantage with lock down techniques is that considerable analysis of the code that is to be run is required to determine which items should and which items should not be locked down within the cache. In order to use lock down to its best effect, each piece of code has to be manually tuned to the cache and system upon which it operates.

Viewed from one aspect the present invention provides apparatus for processing data, said apparatus comprising:

a cache memory;

a storage control circuit for controlling storage of a new word within said cache memory following a cache miss resulting from a cache request to said new word from one of a plurality of request sources;

wherein said storage control circuit is responsive to a programmable partition

setting to divide said cache memory into a plurality of portions each with a storage capacity controlled by said programmable partition setting and said storage control circuit selects in which of said plurality of portions to store said new word in dependence upon which of said plurality of request sources requested said new word.

The invention provides a cache memory that may be programmably partitioned between words requested by different request sources. For example, the instruction pipeline within a central processing unit may act as one request source for instruction words and the register bank within a central processing unit may act as another request source for data words. The cache requests may be read requests in a read allocate cache, write requests in a write allocate cache or both. The division of the available cache memory capacity between instruction words and data words is not fixed by the manufacture of the system and so can be varied to suit the particular task being performed. Compared with the analysis required to effectively use lock down, the determination of the best programmable partition setting is relatively straightforward since the software may simply be run at different settings and the overall performance observed without having to understand or track in detail which words were and were not cached at a particular time. Furthermore, the programmable partition setting may be changed (or switched off completely allowing a unified cache mode) during operation giving an additional degree of sophistication if required. For example, should a program be entering a portion of digital signal processing that is highly data intensive but relatively instruction unintensive, then the partition between data and instruction cache storage can be moved to allow more data cache storage.

It would be possible to design a system such that the cache requests were directed only to the appropriate portion of the cache memory depending upon the request source. However, in preferred embodiments of the invention said cache request searches all of said portions for said new word.

This feature allows the partitioning of the cache to be changed whilst operation is occurring knowing that cached data that is now in the "wrong" portion due to the change will still be found and written back to main memory when replaced so avoiding consistency problems. It has been found that the invention may be effectively implemented by modifying the replacement mechanisms such that data is only ever written into its allocated portion but that the advantages of only searching

for data within the appropriate allocated portion and flushing data (a very time consuming operation when many slow external memory accesses have to be made, particularly for a write back cache) when there is a change in partition are outweighed by the complexity and cost of adapting the system to achieve this. Furthermore, this simple implementation copes with in-line data (without having to store two copies) and self modifying code.

As discussed above, the invention may be particularly useful in embodiments in which the request sources include a data request source and an instruction request source within a central processing unit, which often share an access port. Other examples of systems in which the invention is particularly useful would be those in which a cache memory is partitioned between words required by a central processing unit and words required by a coprocessor or partitioned between different program tasks in a multi-tasking system.

Another preferred feature of the invention is that said storage control circuit selects which currently stored word within said selected portion of said cache memory to overwrite with said new word using independent algorithms for each of said plurality of portions.

The partitioning of the cache allows for the possibility of using different replacement algorithms within the different portions, e.g. random replacement, least recently used, not most recently used, cyclic.

Viewed from another aspect the present invention provides a method of processing data, said method comprising the steps of:

storing words within a cache memory;

controlling, with a storage control circuit, storage of a new word within said cache memory following a cache miss resulting from a cache request to said new word from one of a plurality of request sources;

wherein said storage control circuit is responsive to a programmable partition setting to divide said cache memory into a plurality of portions each with a storage capacity controlled by said programmable partition setting and said storage control circuit selects in which of said plurality of portions to store said new word in dependence upon which of said plurality of request sources requested said new word.

Embodiments of the invention will now be described, by way of example only,

with reference to the accompanying drawings in which:

Figure 1 illustrates a data processing system incorporating a cache memory with a programmable partition between request sources;

Figure 2 illustrates the operation of the system of Figure 1 when reading a word from the cache;

Figure 3 illustrates the operation of the system of Figure 1 when writing a data word to the cache;

Figure 4 illustrates the operation of the system of Figure 1 when writing an instruction word to the cache;

Figure 5 illustrates a first partition of the cache of Figure 1;

Figure 6 illustrates a second partition of the cache of Figure 1;

Figure 7 illustrates a third partition of the cache of Figure 1;

Figure 8 illustrates a system incorporating a central processing unit and coprocessor with a cache partitioned between these two sources; and

Figure 9 illustrates a system incorporating a central processing unit operating in a multi-tasking mode with an instruction cache partitioned between tasks and a data cache partitioned between tasks.

Figure 1 illustrates a system incorporating a cache memory 2 operating in conjunction with a central processing unit 4. The cache memory 2 is composed of four banks of memory (B00, B01, B10 and B11) each with an associated TAG portion T. The cache memory 2 is configured as a 4-way associative (TAG based) cache memory with one word per line and using a random replacement algorithm. An address bus 6 and a data bus 8 connect the central processing unit 4 and the cache memory 2. Data being written to or read from the cache memory 2 is asserted on the data bus 8 with the address with which it is associated being asserted on address bus 6 such that the correct row within the cache memory 4 can be identified and the TAG for the word written, if appropriate. A storage control circuit 10 is provided that comprises an incrementing counter 12, a decrementing counter 14, an incrementing counter comparator 16, a decrementing counter comparator 18, a partition setting register 20, a multiplexer 22 and a decoder 24. A clock signal clk is supplied to both the incrementing counter 12 and the decrementing counter 14 to trigger incrementing and decrementing respectively within a range of values defined by the programmable

partition setting PartVal stored within the partition setting register 20. The clock signal clk is derived from a linear feedback shift register using feedback to generate a random sequence of bits that triggers the decrementing counter 14 and the incrementing counter 16 such that they effectively change randomly. This randomness provides resistance to pathological replacement conditions arising in use. When the counters 12, 14 are sampled to determine the bank within which a new word should be written. the value read out appears essentially random within the range within which it varies.

The incrementing counter 12 increments it value with each pulse of the clock signal clk. This value is then supplied to the incrementing counter comparator 16 where it is tested to see if it has reached the maximum value of 11. When this condition is met incrementing counter is reset by a signal R to load the value PartVal stored in the partition setting register 20 plus 1. The decrementing counter 14 operates in a similar manner except that with each clock signal pulse clk, its count decrements and when the decrementing counter comparator 18 determines its value is 00, then the decrementing counter is reset by a signal R to load the value PartVal stored in the partition setting register.

A multiplexer 22 selects one of the contents of the incrementing counter 12 and the decrementing counter 14 and supplies it to a decoder 24. The multiplexer 22 is switched by a signal I/D from the central processing unit 4 that indicates the request source that is triggering the new line to be written into the cache memory 2. If the I/D signal indicates that the instruction pipeline within the central processing unit core 4 was the source of the cache miss, then the I/D value is set to I and the incrementing counter value is selected by the multiplexer 22 and decoded by the decoder 24. Conversely, if the I/D signal indicates that a register load of a data word was the source of the cache miss for the new word that is now being stored within the cache memory 2, then the I/D signal is set to D, the multiplexer 22 selects the contents of the decrementing counter 12 to be passed to the decoder 24.

The programmable partition setting PartVal is loaded into the programmable setting register 20 from the central processing unit 4 as a register load under program control. At the start of a particular software program to be executed, or indeed within that program, then the programmable setting register 20 may be loaded with the

desired value. In a system with a coprocessor, the partition value register 20 may be a coprocessor register that is loaded from a central processing unit register using a move coprocessor register instruction.

In the case of a linefetch following a cache miss, as distinct from a processor write to the cache memory 2, a write enable signal WE issued from the central processing unit 4 to the decoder 24 serves to only enable writes to the cache memory 2 via the appropriate write enable line to an individual bank. In this case a linefetch following a cache miss to a cacheable area of memory is cause be a processor read access. Processor writes are written directly into the cache memory 2 by a different mechanism.

Figure 2 illustrates the operation of the system of Figure 1 to read a word from the cache memory 2. The write enable signal WE is disabled. In this case, the word to be read is a data word and so that I/D signal is set to D. The cache memory 2 in this case is partitioned to hold a single bank of instruction words I and three banks of data words D. The cache request is passed to all portions of the cache memory 2 such that the TAGs for each of the banks of a given row is compared with the higher order bits of the address on the address bus 6 to determine whether any of the cache locations is storing the required word. In this case, bank B10 is storing the word resulting in a TAG match and the required data word being asserted upon the data bus 8 and return to the central processing unit 4. This embodiment is a TAG based primary cache, although the technique is equally applicable to CAM based caches and secondary caches.

Figure 3 illustrates the operation of the system of Figure 1 when writing a data word to the cache memory 2. The writing of this data word to the cache memory 2 is subsequent to a cache miss from a load register instruction resulting in the data word having to be fetched from the external memory. As the data word is asserted upon the data bus 8 and its address asserted upon the address bus 6, the write enable signal WE is asserted. The I/D signal indicates that a data word D is being written and so the multiplexer 22 selects the current output of the decrementing counter 14 (as indicated by a *) and supplies this to the decoder 24. The current output of the decrementing counter 14 is 01 indicating that bank B01 should be used to store the new data word. The programmable partition setting previously loaded into the

programmable setting register 20 is 10 indicating that the first three banks of the cache memory 2 should be used for data and only the top bank should be used for instructions. The content of the decrementing counter 14 thus follows the sequence 10, 01, 00, 10, ... whilst the incrementing counter 12 provides a constant output of 11. The decoder 24 serves to decode the two bit value fed to it from the decrementing counter 14 to write enable a single one of the banks of the cache memory 2 via the bank enable line indicated by a *.

Figure 4 illustrates the operation of the system of Figure 1 when storing an instruction word in the cache memory 2 following a cache miss. This operation is similar to that illustrated in Figure 3 except that the I/D signal now indicates an instruction word I so causing the multiplexer 22 to select the output of the incrementing counter 12 to be decoded by the decoder 24. Given the setting of the programmable partition setting, the single instruction bank B11 is write enabled for the storage of the new write word.

Figure 5 schematically illustrates the partition of the cache memory 2 of the system of Figure 1 and the manner in which the replacement bank selection is varied. In this case, a single bank is always selected for instruction words and one of the three possible banks is selected for data words. The decrementing counter 14 is responsible for which of the banks is selected for a data word. Since writes to the cache occur in no fixed relationship to the clocking of the decrementing counter 14 and the incrementing counter 12, the sampling of this counter produces an effectively random selection of one of the three banks for the data word. Since with this setting of the programmable partition setting, only a single bank is available for instruction words, this bank is continuously selected for instruction word writes.

Figure 6 illustrates the arrangement when two banks each of the cache memory 2 are allocated for data words and instructions words.

Figure 7 illustrates the situation in which three banks are allocated for instruction words and a single bank is allocated for data words.

Figure 8 illustrates a second embodiment of the invention. In this case the system comprises a central processing unit 26 and a coprocessor 28. The system has a data cache 30 and a separate instruction cache 32. The coprocessor 28 is passed instructions by the central processing unit 26 and so does not require any direct access

to the instruction cache 32. However, the coprocessor 28 and the central processing unit 26 both have direct access to the data cache 30. According to this embodiment, the data cache 30 is partitioned between central processing unit data requested by the central processing unit 26 and coprocessor data requested by the coprocessor 28. The programmable partition setting illustrated shows a large proportion of the storage capacity of the data cache 30 is allocated to coprocessor data. A bus controller 34 controls the routing of words to and from the data cache 30 and the instruction cache 32.

Figure 9 illustrates a further embodiment of the invention. In this case, a central processing unit 36 is provided with a data cache 38 and an instruction cache 40. The central processing unit 36 is operating in a multi-tasking role using three quasi-independent tasks Task1, Task2 and Task3. The data cache 38 is partitioned into portions each corresponding to a respective one of the tasks being performed by the central processing unit 36. The instruction cache 40 is similarly partitioned between instructions corresponding to the various tasks. The relative proportions of the available capacity allocated to each task need not be the same between the instruction words and the data words for that task. A bus controller 42 controls the routing of data to and from the data cache 38 and the instruction cache 40.

CLAIMS

1. Apparatus for processing data, said apparatus comprising:

a cache memory;

a storage control circuit for controlling storage of a new word within said cache memory following a cache miss resulting from a cache request to said new word from one of a plurality of request sources;

wherein said storage control circuit is responsive to a programmable partition setting to divide said cache memory into a plurality of portions each with a storage capacity controlled by said programmable partition setting and said storage control circuit selects in which of said plurality of portions to store said new word in dependence upon which of said plurality of request sources requested said new word.

2. Apparatus as claimed in claim 1, wherein said cache request searches all of said portions for said new word.

3. Apparatus as claimed in claim 2, wherein said cache memory is an N-way associative cache memory, where N is an integer value greater than 1, and each of said plurality of portions has a storage capacity selectable in steps of 1/N of the total cache memory storage capacity.

4. Apparatus as claimed in any one of claims 1, 2 and 3, comprising a central processing unit having a data request source for requesting new data words and an instruction request source for requesting new instruction words.

5. Apparatus as claimed in claim 4, wherein said cache memory has a data word portion and an instruction word portion, said programmable partition setting controlling division of capacity of said cache memory between data words and instruction words.

6. Apparatus as claimed in any one of the preceding claims, wherein said storage control circuit selects which currently stored word within said selected portion of said

cache memory to overwrite with said new word using independent algorithms for each of said plurality of portions.

7. A method of processing data, said method comprising the steps of:

storing words within a cache memory;

controlling, with a storage control circuit, storage of a new word within said cache memory following a cache miss resulting from a cache request for said new word from one of a plurality of request sources;

wherein said storage control circuit is responsive to a programmable partition setting to divide said cache memory into a plurality of portions each with a storage capacity controlled by said programmable partition setting and said storage control circuit selects in which of said plurality of portions to store said new word in dependence upon which of said plurality of request sources requested said new word.

8. Apparatus for processing data substantially as hereinbefore described with reference to the accompanying drawings.

9. A method of processing data substantially as hereinbefore described with reference to the accompanying drawings.

The Patent Office

12

Application No:	GB 9607044.6	**Examiner:**	B G Western
Claims searched:	1 to 9	**Date of search:**	5 July 1996

Patents Act 1977
Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:

UK Cl (Ed.O): G4A AMC

Int Cl (Ed.6): G06F 12/08

Other: On-line : WPI, Inspec, Computer

Documents considered to be relevant:

Category	Identity of document and relevant passage			Relevant to claims
X	GB-2292822-A	(Hewlett-Packard)	See whole document	1,2,4,5,7
X	GB-2250114-A	(Mitsubishi Denki)	N.b. pages 7-9	1,3,4,5,7
X	GB-2214336-A	(Mitsubishi Denki)	N.b. pages 7-9	1,3,4,5,7
X	EP-0481616-A2	(IBM)	N.b. page 7	1,4,5,6,7
X	EP-0466265-A1	(Philips)	N.b. pages 4-7	1,4,5,6,7
X	EP-0442474-A2	(Sanyo)	N.b. column 3	1,4,7
X	EP-0075714-A2	(Siemens)	N.b. page 3 lines 21-32	1,4,5,7
X	US-5434992-A	(Mattson)	N.b. columns 1-5	1,4,5,7

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
		E	Patent document published on or after, but with priority date earlier than, the filing date of this application.
&	Member of the same patent family		